FIG. 1

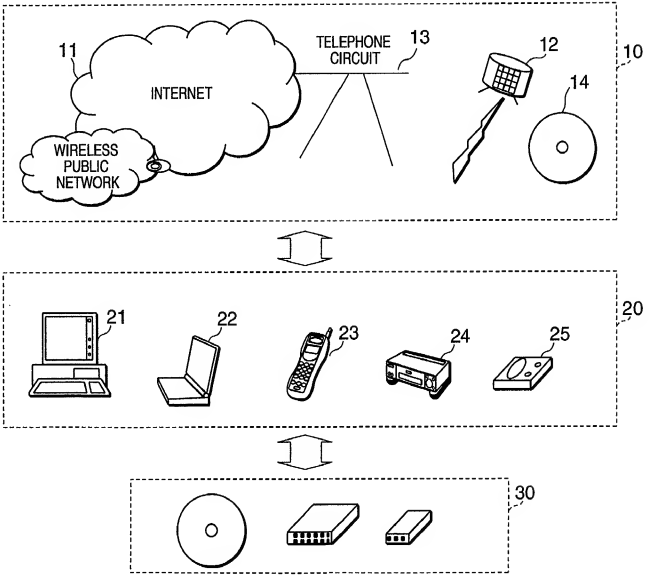


FIG. 2

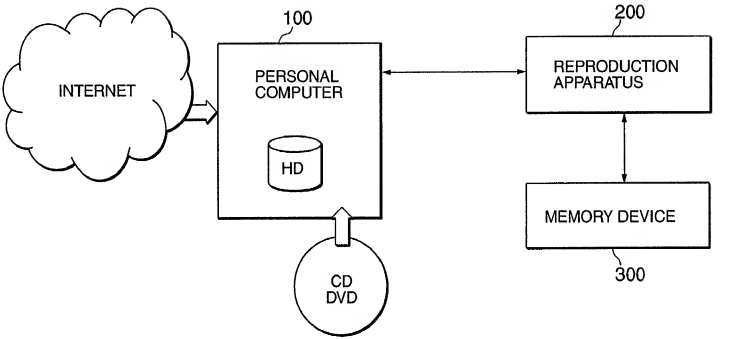


FIG. 3

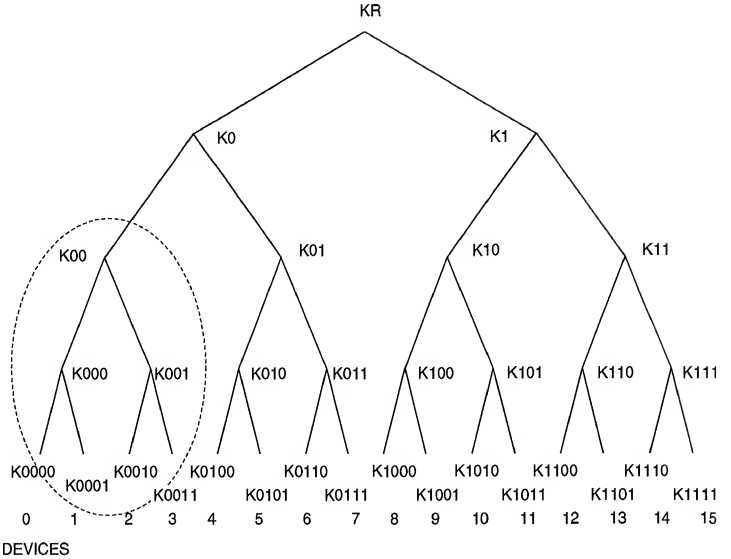


FIG. 4

(A)

EKB (ENABLING KEY BLOCK) EXAMPLE 1
DELIVERS NODE KEYS OF VERSION (t) TO DEVICES 0, 1,AND 2

VERSION : t	
INDEX	ENCIPHERING KEY
0	Enc(K(t)0, K(t)R)
00	Enc(K(t)00, K(t)0)
000	Enc(K000, K(t)00)
001	Enc(K(t)001, K(t)00)
0010	Enc(K0010, K(t)001)

(B)

EKB (ENABLING KEY BLOCK) EXAMPLE 2
DELIVER NODE KEY OF VERSION (t) TO DEVICES 0, 1, AND 2

VERSION : t	
INDEX	ENCIPHERING KEY
000	Enc(K000, K(t)00)
001	Enc(K(t)001, K(t)00)
0010	Enc(K0010, K(t)001)

FIG. 5

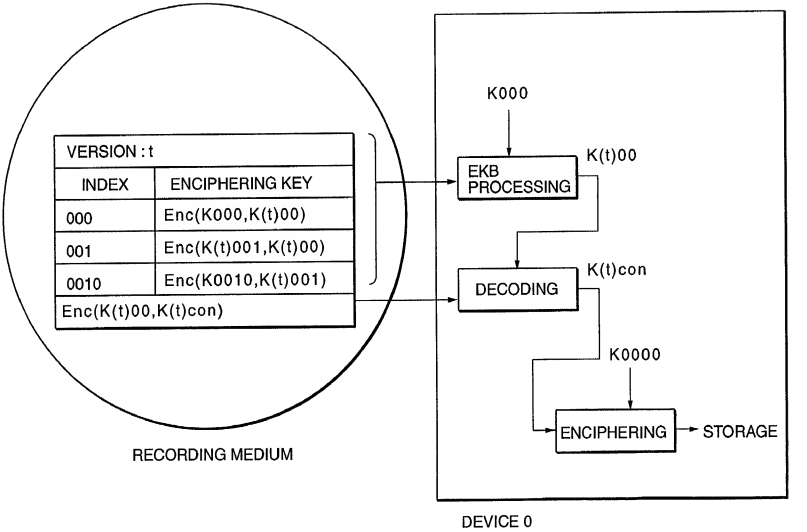


FIG. 6

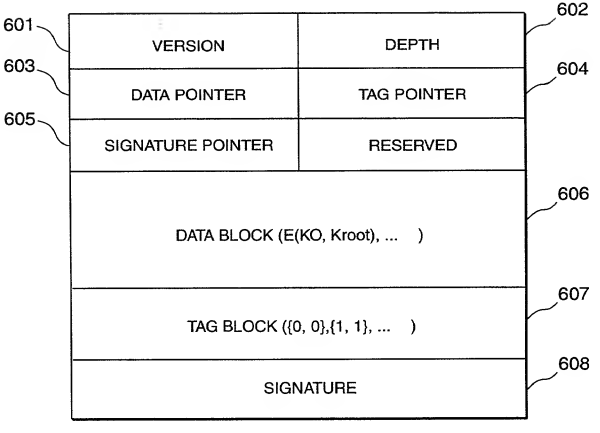
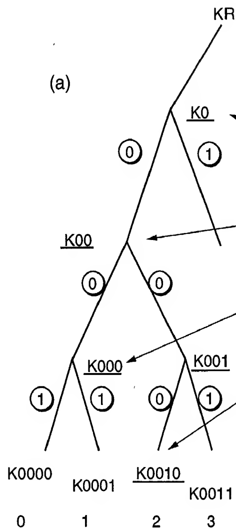

601
603
605
602
604
606
607
608
VERSION
DEPTH
DATA POINTER
TAG POINTER
SIGNATURE POINTER
RESERVED
DATA BLOCK (E(KO, Kroot), ...)
TAG BLOCK ({0, 0},{1, 1}, ...)
SIGNATURE

FIG. 7

(a)



(b)

EKB (ENABLING KEY BLOCK)

DELIVER NODE KEY OF VERSION (t) TO DEVICES 0, 1, AND 2

TOP-NODE ADDRESS : KR	
DATA (ENCIPHERED KEY)	TAG
Enc(K(t)0,K(t)R)	{0,1}
Enc(K(t)00,K(t)0)	{0,0}
Enc(K000,K(t)oo)	{1,1}
Enc(K(t)001,K(t)00)	{0,1}
Enc(K(t)0010,K(t)001)	{1,1}

{L-TAG, R-TAG}
IF DATA IS PRESENT IN RESPECTIVE DIRECTION(RIGHT AND LEFT), TAG VALUE IS 0 AND
IF NO DATA IS PRESENT, TAG VALUE IS 1.

(c)

DATA : Enc(K(t)0,K(t)R),Enc(K(t)00,K(t)0), ...

TAG: {0,1},{0,0},{1,1} ...

FIG. 8

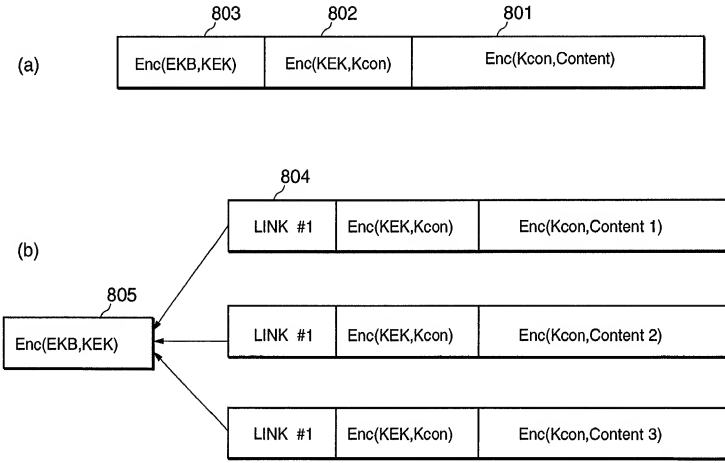


FIG. 9

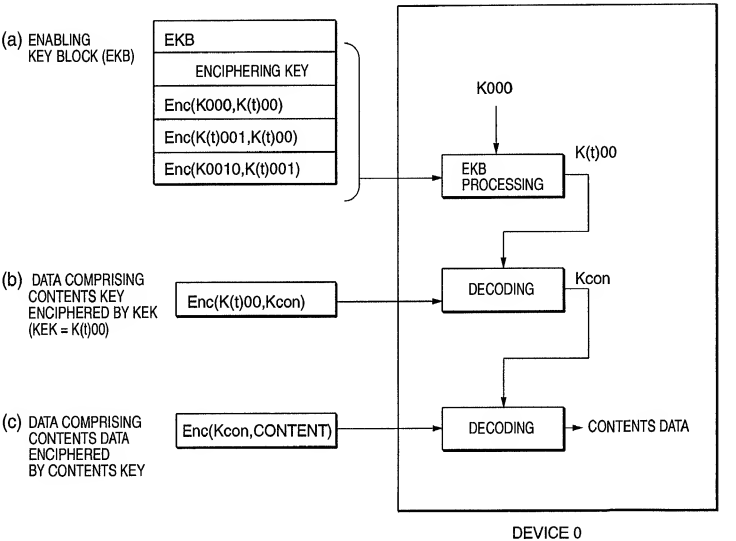
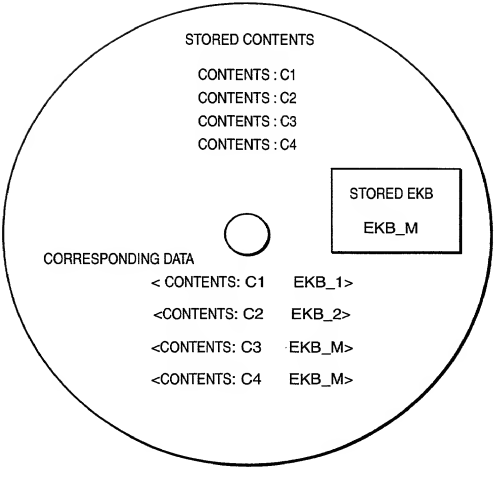

(a) ENABLING KEY BLOCK (EKB)
EKB
ENCIPHERING KEY
Enc(K000,K(t)00)
Enc(K(t)001,K(t)00)
Enc(K0010,K(t)001)
K000
EKB PROCESSING
K(t)00
(b) DATA COMPRISING CONTENTS KEY ENCIPHERED BY KEK (KEK = K(t)00)
Enc(K(t)00,Kcon)
DECODING
Kcon
(c) DATA COMPRISING CONTENTS DATA ENCIPHERED BY CONTENTS KEY
Enc(Kcon,CONTENT)
DECODING
CONTENTS DATA
DEVICE 0

FIG. 10



RECORDING MEDIUM

FIG. 11

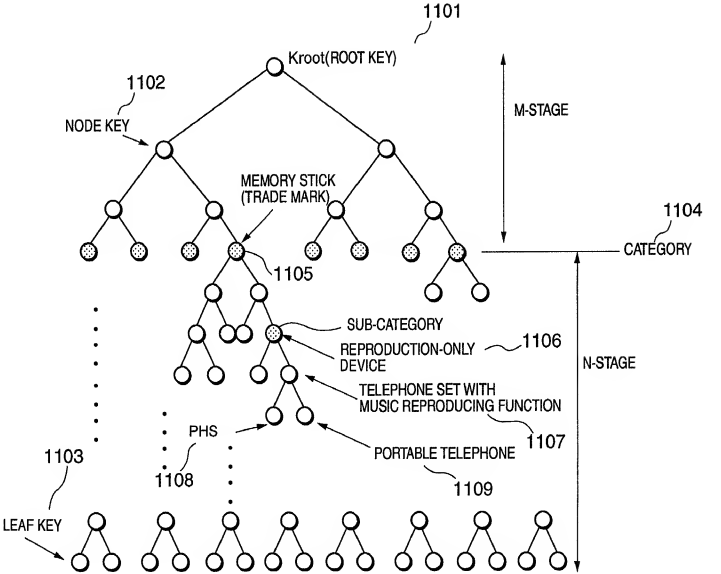

1101
Kroot(ROOT KEY)
1102
NODE KEY
M-STAGE
MEMORY STICK
(TRADE MARK)
1104
CATEGORY
1105
SUB-CATEGORY
REPRODUCTION-ONLY
DEVICE
1106
N-STAGE
TELEPHONE SET WITH
MUSIC REPRODUCING FUNCTION
PHS
1107
PORTABLE TELEPHONE
1103
1108
1109
LEAF KEY

FIG. 12

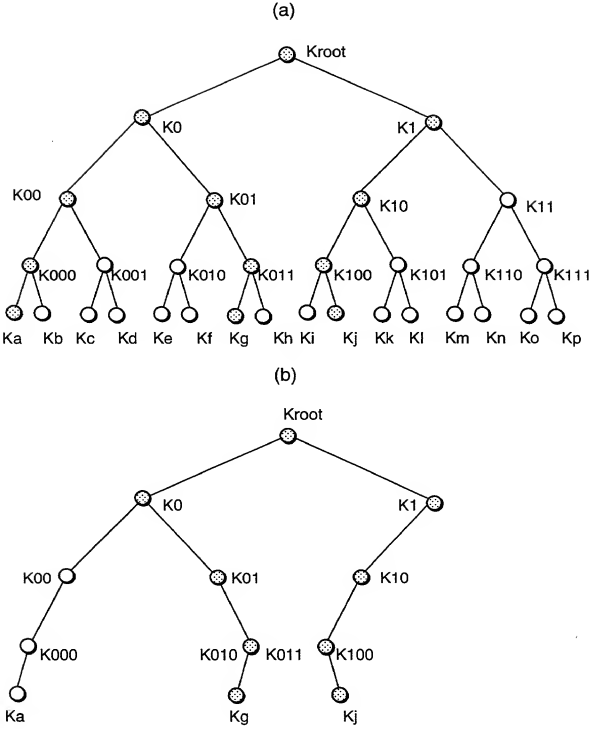
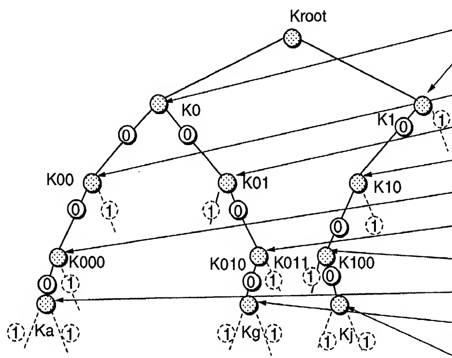

(a)
Kroot
K0
K1
K00
K01
K10
K11
K000
K001
K010
K011
K100
K101
K110
K111
Ka
Kb
Kc
Kd
Ke
Kf
Kg
Kh
Ki
Kj
Kk
Kl
Km
Kn
Ko
Kp
(b)
Kroot
K0
K1
K00
K01
K10
K000
K010
K011
K100
Ka
Kg
Kj

FIG. 13

(a)



(b)

TAG	DATA(ENCIPHERING KEY)
{0, 0}	Enc(K(t)0,K(t)root)
{0, 1}	Enc(K(t)1,K(t)root)
{0, 1}	Enc(K(t)00,K(t)0)
{1, 0}	Enc(K(t)01,K(t)0)
{0, 1}	Enc(K(t)10,K(t)1)
{0, 1}	Enc(K(t)000,K(t)00)
{0, 1}	Enc(K(t)011,K(t)01)
{1, 0}	Enc(K(t)100,K(t)10)
{1, 1}	Enc(Ka,K(t)000)
{1, 1}	Enc(Kg,K(t)011)
{1, 1}	Enc(Kj,K(t)100)
Enc(K(t)root, K(t)con)	

FIG. 14

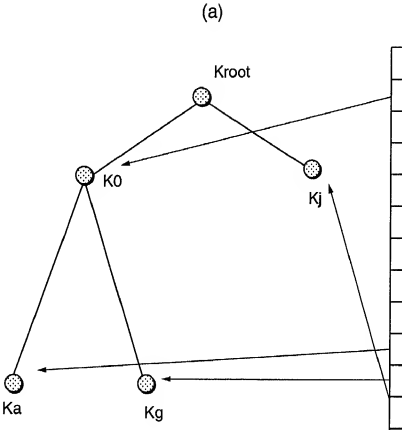

(a)
(b)
Kroot
K0
Kj
Ka
Kg
TAG
DATA(ENCIPHERING KEY)
{0, 0, 1}
Enc(K(t)0,K(t)root)
{0, 1, 0}
–
{0, 1, 0}
–
{1, 0, 0}
–
{0, 1, 0}
–
{0, 1, 0}
–
{0, 1, 0}
–
{1, 0, 0}
–
{1, 1, 1}
Enc(Ka,K(t)0)
{1, 1, 1}
Enc(Kg,K(t)0)
{1, 1, 1}
Enc(Kj,K(t)root)
Enc(K(t)root, K(t)con)

FIG. 15

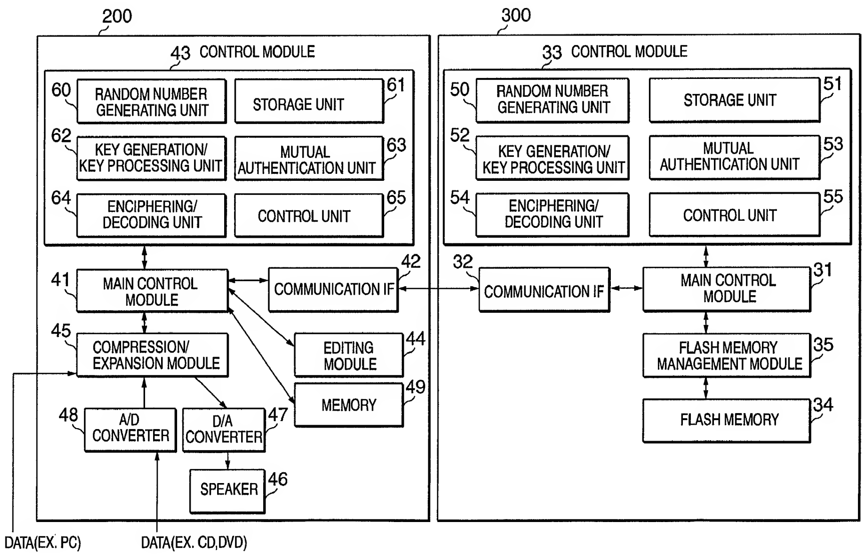

200
300
43 CONTROL MODULE
60 RANDOM NUMBER GENERATING UNIT
61 STORAGE UNIT
62 KEY GENERATION/ KEY PROCESSING UNIT
63 MUTUAL AUTHENTICATION UNIT
64 ENCIPHERING/ DECODING UNIT
65 CONTROL UNIT
41 MAIN CONTROL MODULE
42 COMMUNICATION IF
45 COMPRESSION/ EXPANSION MODULE
44 EDITING MODULE
49 MEMORY
48 A/D CONVERTER
47 D/A CONVERTER
46 SPEAKER
DATA(EX. PC)
DATA(EX. CD,DVD)
33 CONTROL MODULE
50 RANDOM NUMBER GENERATING UNIT
51 STORAGE UNIT
52 KEY GENERATION/ KEY PROCESSING UNIT
53 MUTUAL AUTHENTICATION UNIT
54 ENCIPHERING/ DECODING UNIT
55 CONTROL UNIT
32 COMMUNICATION IF
31 MAIN CONTROL MODULE
35 FLASH MEMORY MANAGEMENT MODULE
34 FLASH MEMORY

FIG. 16

DATA STORED IN A STORAGE UNIT OF A MEMORY DEVICE

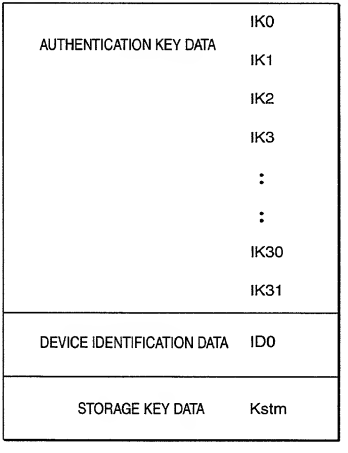


FIG. 17

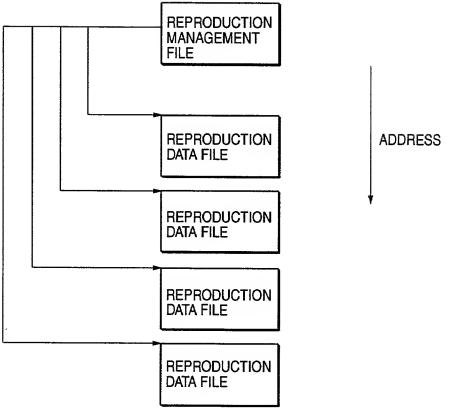

REPRODUCTION MANAGEMENT FILE
REPRODUCTION DATA FILE
REPRODUCTION DATA FILE
REPRODUCTION DATA FILE
REPRODUCTION DATA FILE
ADDRESS

FIG. 18

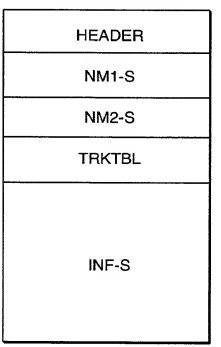

REPRODUCTION MANAGEMENT FILE
HEADER
NM1-S
NM2-S
TRKTBL
INF-S

FIG. 19

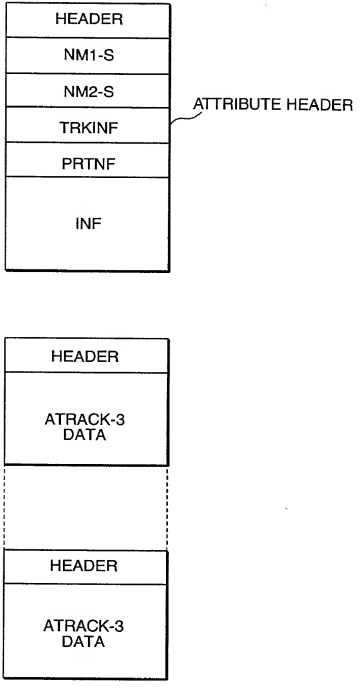

HEADER
NM1-S
NM2-S
TRKINF
PRTNF
INF
ATTRIBUTE HEADER
HEADER
ATRACK-3 DATA
HEADER
ATRACK-3 DATA

FIG. 20

REPRODUCTION MANAGEMENT FILE

A

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0x0000	BLKID-TLO				RESERVED		MCODE		REVISION				RESERVED			
0x0010	SN1C+L		SN2C+L		SINFSIZE		T-TRK		VerNo.		RESERVED					

B

0x0020	NM1-S(256)							
0x0120	NM2-S(512)							
0x0310								
0x0320	RESERVED(4)		EKB VERSION		E(Kstm,Kcon)			
0x0330	E(KEKn,Kcon)				c_MAC[0]			
0x0340	RESERVED(8)				RESERVED(3)	MGR	S-YMDhms	
0x0350	TRK-001	TRK-002	TRK-003	TRK-004	TRK-005	TRK-006	TRK-007	TRK-008
0x0360	TRK-009	TRK-010	TRK-011	TRK-012	TRK-013	TRK-014	TRK-015	TRK-016
0x0660	TRK-393	TRK-394	TRK-395	TRK-396	TRK-397	TRK-398	TRK-399	TRK-400
0x0670	INF-S(14720)							
0x3FFF	BLKID-TLO		RESERVED	MCODE	REVISION		RESERVED	

C

0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
INF	0X00	ID	0X00	SIZE		MCODE		C+L		RESERVED	DATA VARIABLE LENGTH				

FIG. 21

ATRACK-3 DATA FILE

	0 1 2 3 4 5 6 7	8 9 A B	C D E F
0x0000	BLKID-HDO / RESERVED / MCODE	RESERVED	BLOCK SERIAL
0x0010	N1C+L / N2C+L / INFSIZE / T-PRT	T-SU	INX / XT
0x0020	NM1-S(256)		
0x0120	NM2-S(512)		
0x0310			
0x0320	RESERVED(3) / EKI / EKB VERSION	E(Kstm, Kcon)	
0x0330	E(KEKn. Kcon)	C_MAC[n]	
0x0340	RESERVED(8)	INF_seq#	A / LT / FNo
0x0350	MG(D)SERIAL-nnn(Upper)	MG(D)SERIAL-nnn(LOWER)	
0x0360	CONNUM / YMDhms-S	YMDhms-E	XCC / CT / CC / CN
0x0370	PRTSIZE / PRTKEY		RESERVED(8)
0x0380	CONNUMO	PRTSIZE(0x0388)	PRTKEY
0x0390	RESERVED(8)		CONNUMO
	INF(0x0400)		
0x3FFF	BLKID-HDD / RESERVED / MCODE	RESERVED	BLOCK SERIAL
0x4000	BLKID-A3D / RESERVED / MCODE	CONNUMO	BLOCK SERIAL
0x4010	BLOCKSEED	INTIALIZATION VECTOR	
0x4020	SU-000(NByte=384Byte)		
0x41A0	SU-001(NByte)		
0x4320	SU-002(NByte)		
0x04A0	SU-041(NByte)		
0x7DA0	RESERVED(NByte=208Byte)		
0x7F20	BLK SEED		
0x7FF0	BLKID-A3D / RESERVED / MCODE	CONNUMO	BLOCK SERIAL

FIG. 22

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0x0000	BLKID-HDO			RESERVED		MCODE		RESERVED				BLOCK SERIAL				
0x0010	N1C+L	N2C+L		INFSIZE		T-PRT		T-SU				INX		XT		
0x0020	NM1-S(256)															
0x0120	NM2-S(512)															
0x0310																

FIG. 23

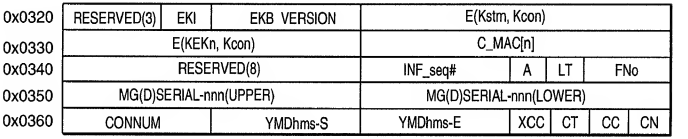

0x0320
RESERVED(3)
EKI
EKB VERSION
E(Kstm, Kcon)
0x0330
E(KEKn, Kcon)
C_MAC[n]
0x0340
RESERVED(8)
INF_seq#
A
LT
FNo
0x0350
MG(D)SERIAL-nnn(UPPER)
MG(D)SERIAL-nnn(LOWER)
0x0360
CONNUM
YMDhms-S
YMDhms-E
XCC
CT
CC
CN

FIG. 24

Bit7 : ATRAC3 Mode 0 : Dual 1 : Joint

Bits 6, 5, 4: N OF 3-Bit CORRESPONDS TO MODE VALUE

N	MODE	TIME	TRANSFER RATE	SU (SOUND UNIT)	Byte
7	HQ	47min	176kbps	31SU	512
6		58min	146kbps	38SU	424
5	EX	64min	132kbps	42SU	384
4	SP	81min	105kbps	53SU	304
3		90min	94kbps	59SU	272
2	LP	128min	66kbps	84SU	192
1	MONO	181min	47kbps	119SU	136
0	MONO	258min	33kbps	169SU	96

Bit3 : RESERVED

Bit2 : DATA DISTINCTION 0 : AUDIO 1 : OTHERS

Bit1 : REPRODUCED SKIP 0 : NORMAL REPRODUCTION 1 : SKIP

Bit0 : EMPHASIS 0 : OFF 1 : ON(50/15 μ SECCOND)

FIG. 25

Bit7 : COPY APPROVAL 0 : COPY INHIBITED 1 : COPY APPROVED

Bit6 : GENERATION (VERSION) 0 : ORIGINAL 1 : BEYOND THE FIRST GENERATION

HCMS Bit5-4 : CONTROL IN RELATION TO HIGH-SPEED DIGITAL COPYING OPERATION

00 : COPY INHIBITED 01 : COPY FOR THE FIRST GENERATION 10 : COPY APPROVED
CHILD WHO IMPLEMENTED COPYING OF THE FIRST GENERATION IS
INHIBITED FROM EXECUTING FURTHER COPYING OPERASTION

Bit3-2 : MAGIC GATE AUTHENTICATION LEVEL

00: LEVEL10(Non-MG) 01 : LEVEL1
02: LEVEL12 11 : RESERVED
02: LEVEL10
THOSE LEVELS OTHER THAN 10 CAN NOT BE DIVIDED NOR COMBINED

Bit1, 0 : RESERVED

FIG. 26

0x0370	PRTSIZE	PRTKEY		RESERVED (8)
0x0380		CONNUMO	PRTSIZE(0x0388)	PRTKEY
0x0390		RESERVED (8)		CONNUMO

FIG. 27

0x4000	BLKID-A3D	RESERVED	MCODE	CONNUMO	BLOCK SERIAL
0x4010	BLOCKSEED			INTIALIZATION VECTOR	
0x4020	SU-000(NByte=384Byte)				

FIG. 28

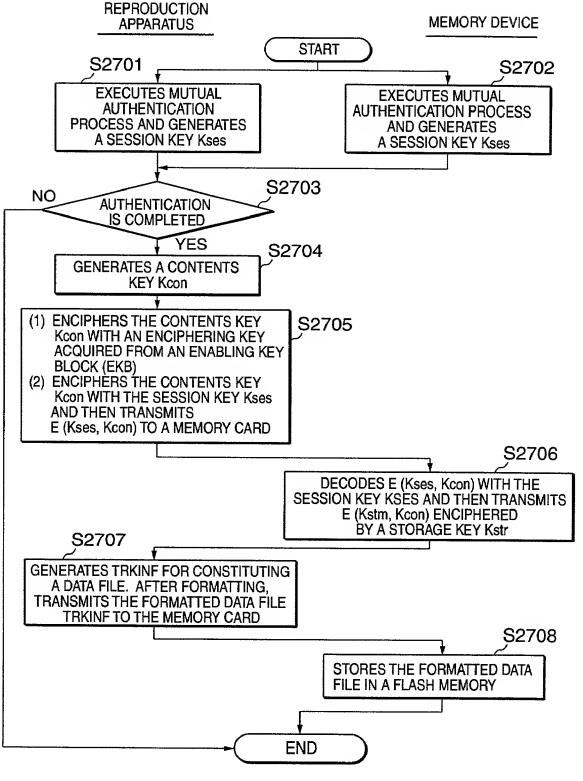
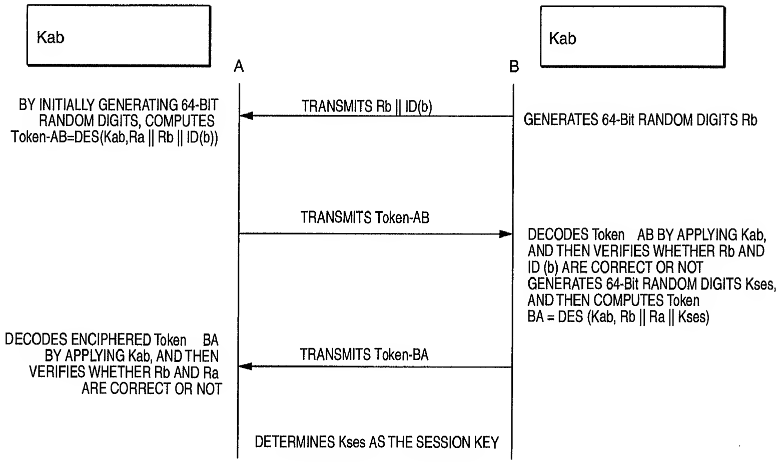

REPRODUCTION APPARATUS
MEMORY DEVICE
START
S2701
EXECUTES MUTUAL AUTHENTICATION PROCESS AND GENERATES A SESSION KEY Kses
S2702
EXECUTES MUTUAL AUTHENTICATION PROCESS AND GENERATES A SESSION KEY Kses
NO
AUTHENTICATION IS COMPLETED
S2703
YES
S2704
GENERATES A CONTENTS KEY Kcon
(1) ENCIPHERS THE CONTENTS KEY Kcon WITH AN ENCIPHERING KEY ACQUIRED FROM AN ENABLING KEY BLOCK (EKB)
(2) ENCIPHERS THE CONTENTS KEY Kcon WITH THE SESSION KEY Kses AND THEN TRANSMITS E (Kses, Kcon) TO A MEMORY CARD
S2705
S2706
DECODES E (Kses, Kcon) WITH THE SESSION KEY KSES AND THEN TRANSMITS E (Kstm, Kcon) ENCIPHERED BY A STORAGE KEY Kstr
S2707
GENERATES TRKINF FOR CONSTITUTING A DATA FILE. AFTER FORMATTING, TRANSMITS THE FORMATTED DATA FILE TRKINF TO THE MEMORY CARD
S2708
STORES THE FORMATTED DATA FILE IN A FLASH MEMORY
END

FIG. 29



MUTUAL AUTHENTICATION FORMAT AND KEY-COMMUNIZING FORMAT VIA UTILIZATION OF THE ISO/IEC9798-2 STANDARD SYMMETRICAL KEY ENCIPHERING ART

FIG. 30

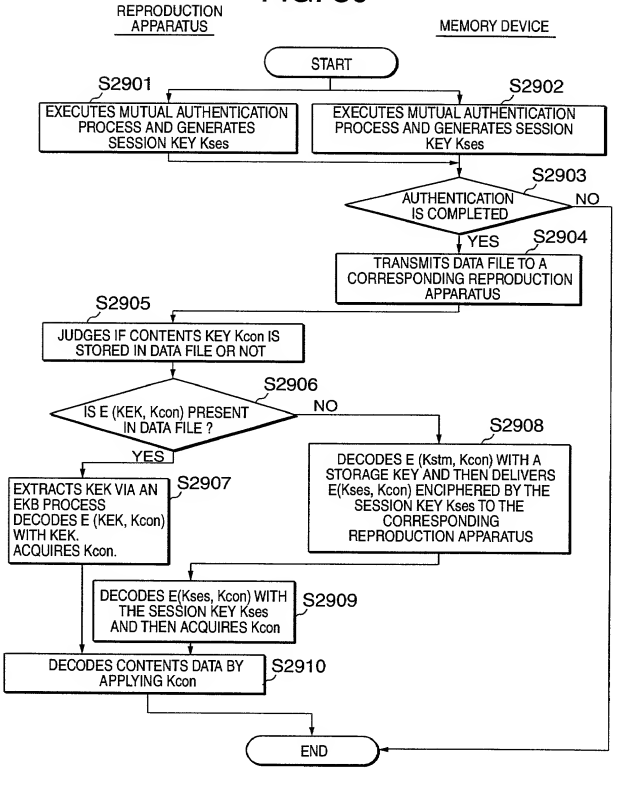

REPRODUCTION APPARATUS
MEMORY DEVICE
START
S2901
EXECUTES MUTUAL AUTHENTICATION PROCESS AND GENERATES SESSION KEY Kses
S2902
EXECUTES MUTUAL AUTHENTICATION PROCESS AND GENERATES SESSION KEY Kses
S2903
AUTHENTICATION IS COMPLETED
NO
YES
S2904
TRANSMITS DATA FILE TO A CORRESPONDING REPRODUCTION APPARATUS
S2905
JUDGES IF CONTENTS KEY Kcon IS STORED IN DATA FILE OR NOT
S2906
IS E (KEK, Kcon) PRESENT IN DATA FILE ?
NO
YES
S2908
DECODES E (Kstm, Kcon) WITH A STORAGE KEY AND THEN DELIVERS E(Kses, Kcon) ENCIPHERED BY THE SESSION KEY Kses TO THE CORRESPONDING REPRODUCTION APPARATUS
S2907
EXTRACTS KEK VIA AN EKB PROCESS DECODES E (KEK, Kcon) WITH KEK. ACQUIRES Kcon.
S2909
DECODES E(Kses, Kcon) WITH THE SESSION KEY Kses AND THEN ACQUIRES Kcon
S2910
DECODES CONTENTS DATA BY APPLYING Kcon
END

FIG. 31

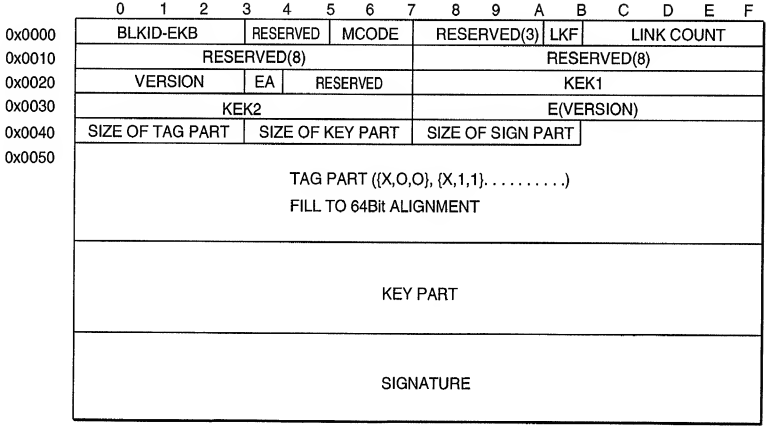

0 1 2 3 4 5 6 7 8 9 A B C D E F
0x0000
BLKID-EKB
RESERVED
MCODE
RESERVED(3)
LKF
LINK COUNT
0x0010
RESERVED(8)
RESERVED(8)
0x0020
VERSION
EA
RESERVED
KEK1
0x0030
KEK2
E(VERSION)
0x0040
SIZE OF TAG PART
SIZE OF KEY PART
SIZE OF SIGN PART
0x0050
TAG PART ({X,O,O}, {X,1,1}. . . . . .)
FILL TO 64Bit ALIGNMENT
KEY PART
SIGNATURE

FIG. 32

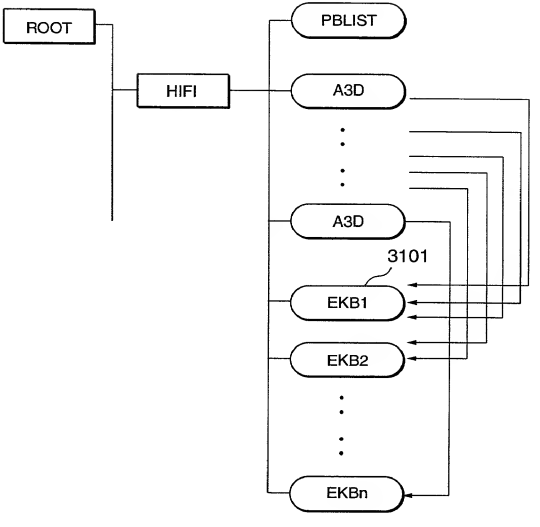


FIG. 33

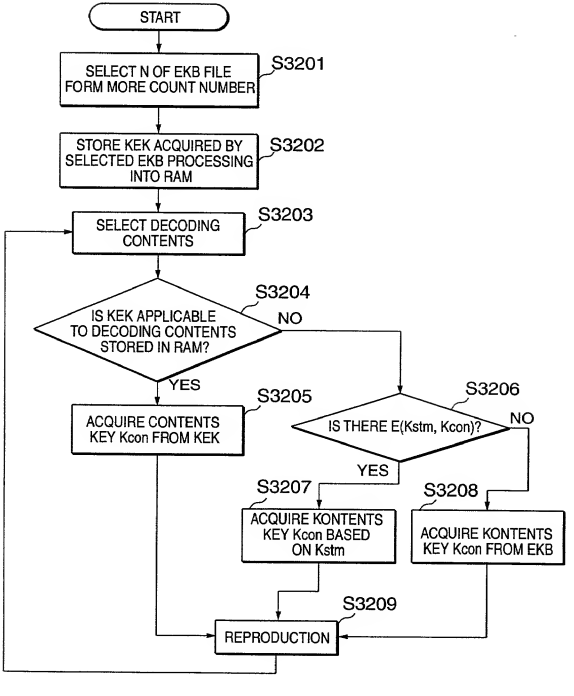

START
SELECT N OF EKB FILE FORM MORE COUNT NUMBER
S3201
STORE KEK ACQUIRED BY SELECTED EKB PROCESSING INTO RAM
S3202
SELECT DECODING CONTENTS
S3203
IS KEK APPLICABLE TO DECODING CONTENTS STORED IN RAM?
S3204
NO
YES
ACQUIRE CONTENTS KEY Kcon FROM KEK
S3205
IS THERE E(Kstm, Kcon)?
S3206
NO
YES
ACQUIRE KONTENTS KEY Kcon BASED ON Kstm
S3207
ACQUIRE KONTENTS KEY Kcon FROM EKB
S3208
REPRODUCTION
S3209

FIG. 34

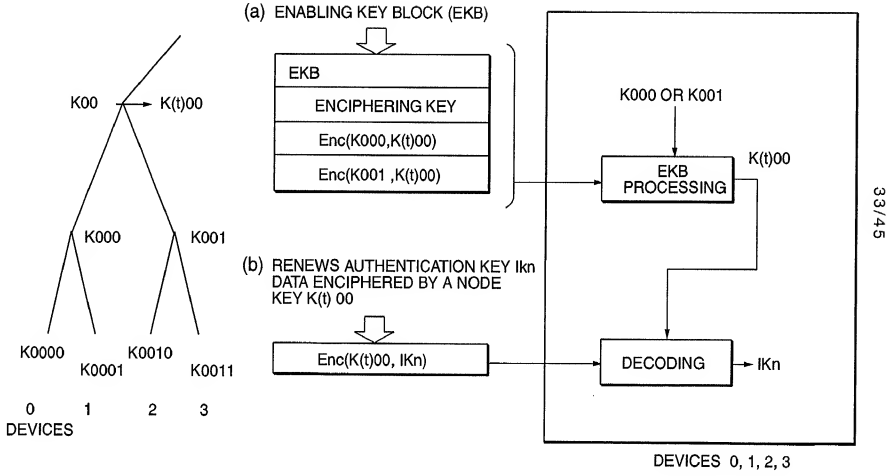

(a) ENABLING KEY BLOCK (EKB)
EKB
ENCIPHERING KEY
Enc(K000,K(t)00)
Enc(K001 ,K(t)00)
K00
K(t)00
K000
K001
K0000
K0001
K0010
K0011
0
1
2
3
DEVICES
K000 OR K001
EKB PROCESSING
K(t)00
(b) RENEWS AUTHENTICATION KEY IKn DATA ENCIPHERED BY A NODE KEY K(t) 00
Enc(K(t)00, IKn)
DECODING
IKn
DEVICES 0, 1, 2, 3

FIG. 35

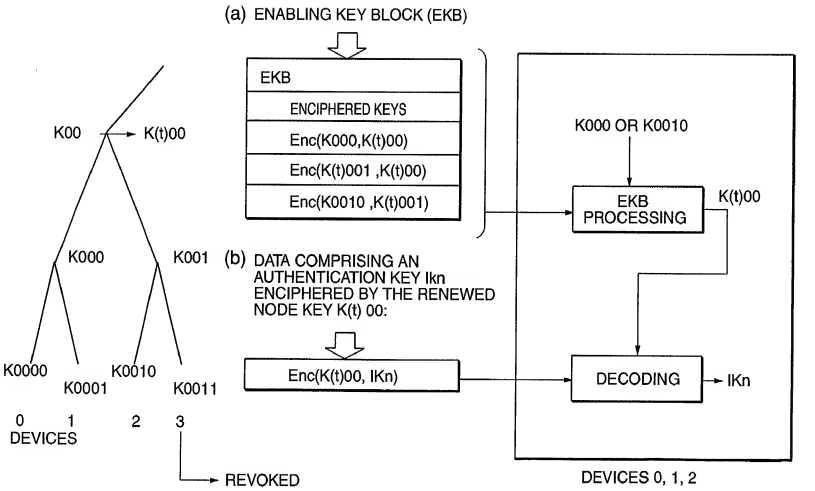

(a) ENABLING KEY BLOCK (EKB)
EKB
ENCIPHERED KEYS
Enc(K000,K(t)00)
Enc(K(t)001 ,K(t)00)
Enc(K0010 ,K(t)001)
K00
K(t)00
K000
K001
K0000
K0001
K0010
K0011
0
1
2
3
DEVICES
REVOKED
(b) DATA COMPRISING AN AUTHENTICATION KEY IKn ENCIPHERED BY THE RENEWED NODE KEY K(t) 00:
Enc(K(t)00, IKn)
K000 OR K0010
EKB PROCESSING
K(t)00
DECODING
IKn
DEVICES 0, 1, 2

FIG. 36

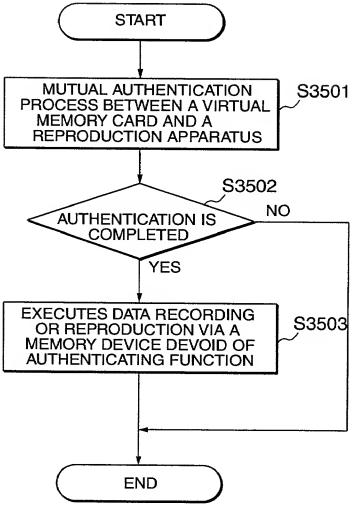
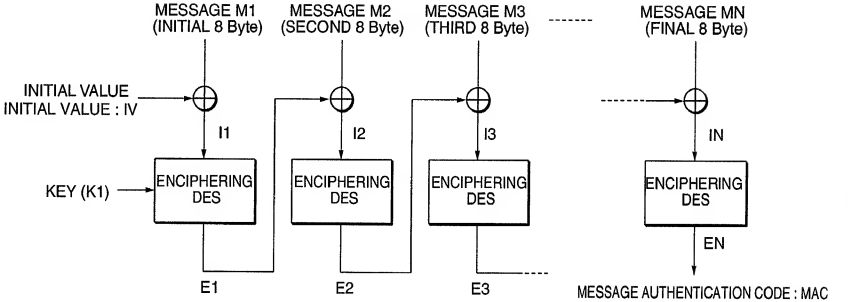

START
MUTUAL AUTHENTICATION PROCESS BETWEEN A VIRTUAL MEMORY CARD AND A REPRODUCTION APPARATUS
S3501
S3502
AUTHENTICATION IS COMPLETED
NO
YES
EXECUTES DATA RECORDING OR REPRODUCTION VIA A MEMORY DEVICE DEVOID OF AUTHENTICATING FUNCTION
S3503
END

FIG. 37



⊕ EXCLUSIVE OR PROCESS (8 Bytes UNIT)

FIG. 38

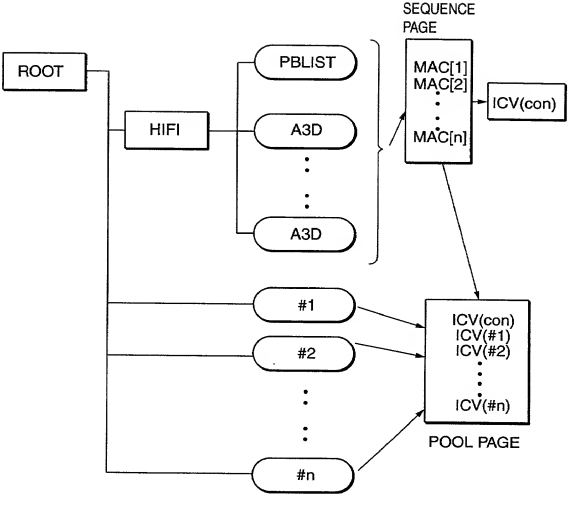

SEQUENCE PAGE
ROOT
PBLIST
HIFI
A3D
A3D
MAC[1]
MAC[2]
MAC[n]
ICV(con)
#1
#2
#n
ICV(con)
ICV(#1)
ICV(#2)
ICV(#n)
POOL PAGE

FIG. 39

SEQUENCE PAGE FORMAT

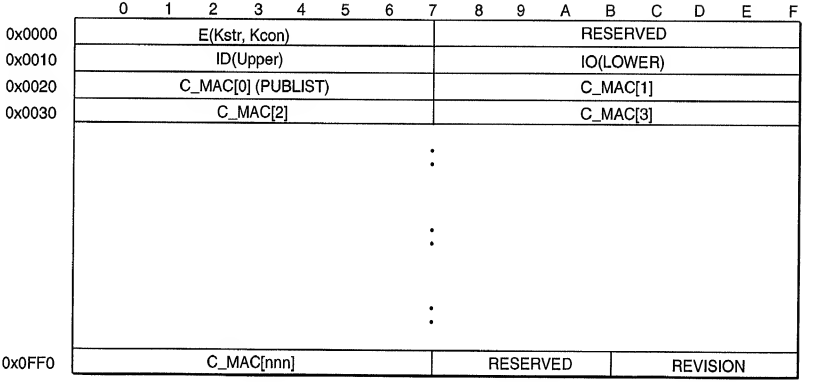


FIG. 40

POOL PAGE FORMAT

	0 1 2 3 4 5 6 7	8 9 A B C D E F
0x0000	#0_REVISION / #0_EKB VERSION	#0_E(KEK, Kicv)
0x0010	#0_E(KEK, Kicv)	ICV0
0x0020	#1_REVISION / #1_EKB VERSION	#1_E(KEK, Kicv)
0x0030	#1_E (KEK,Kicv)	ICV1
	⋮	
0x01E0	#15_REVISION / #15_EKB VERSION	#15_E(KEK, Kicv)
0x01F0	#15_E (KEK,Kicv)	ICV15

FIG. 41

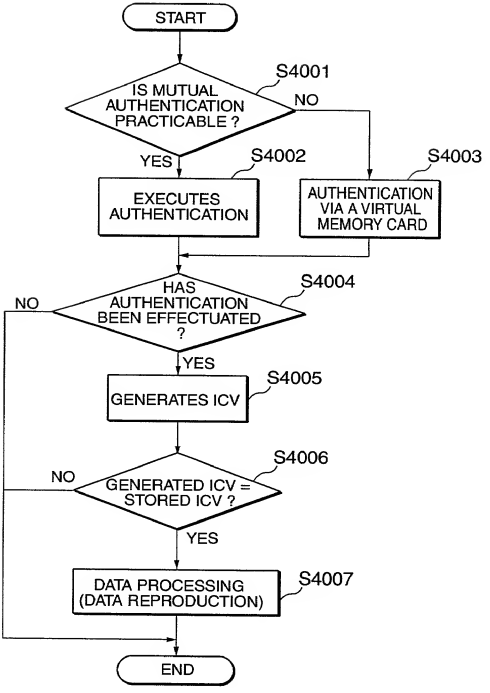

START
S4001
IS MUTUAL AUTHENTICATION PRACTICABLE ?
NO
YES
S4002
EXECUTES AUTHENTICATION
S4003
AUTHENTICATION VIA A VIRTUAL MEMORY CARD
S4004
HAS AUTHENTICATION BEEN EFFECTUATED ?
NO
YES
S4005
GENERATES ICV
S4006
GENERATED ICV = STORED ICV ?
NO
YES
S4007
DATA PROCESSING (DATA REPRODUCTION)
END

FIG. 42

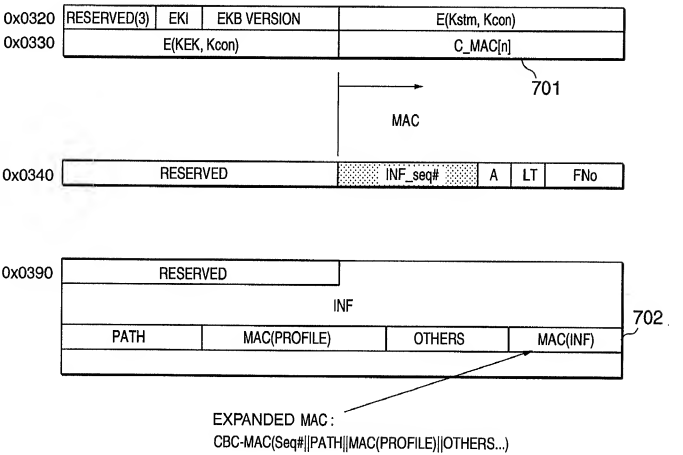


FIG. 43

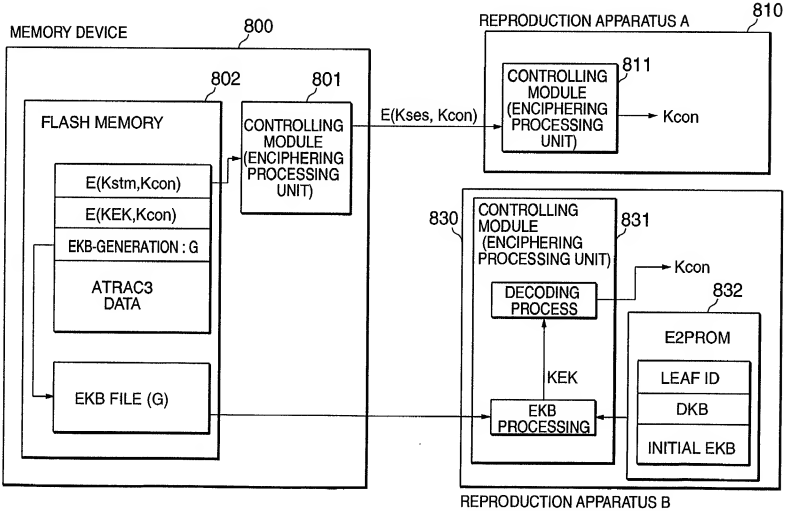


FIG. 44

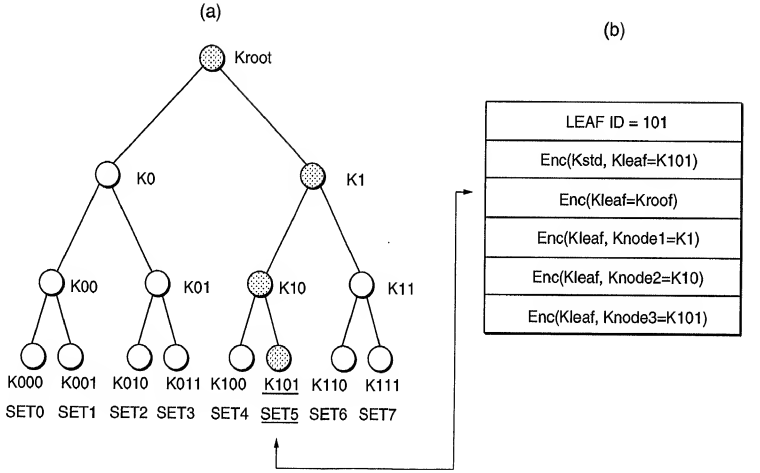
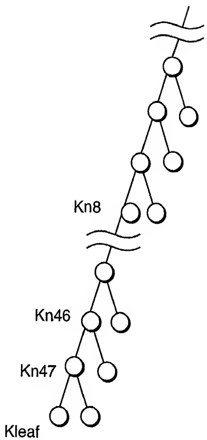

(a)
(b)
Kroot
K0
K1
K00
K01
K10
K11
K000
K001
K010
K011
K100
K101
K110
K111
SET0
SET1
SET2
SET3
SET4
SET5
SET6
SET7
LEAF ID = 101
Enc(Kstd, Kleaf=K101)
Enc(Kleaf=Kroof)
Enc(Kleaf, Knode1=K1)
Enc(Kleaf, Knode2=K10)
Enc(Kleaf, Knode3=K101)

FIG. 45

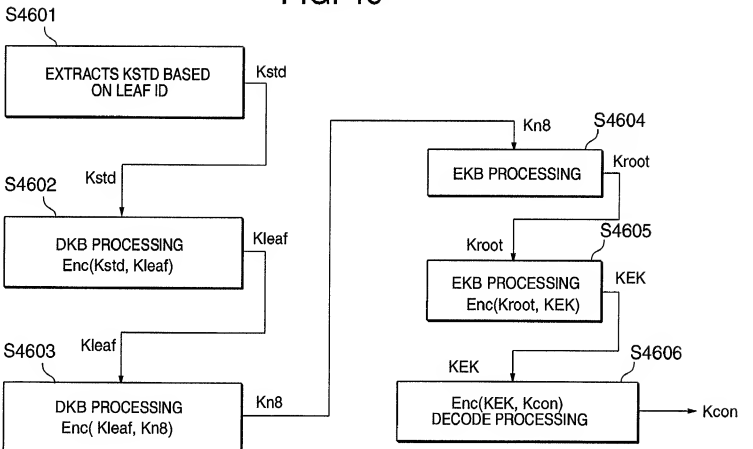
(a)



(b)

LEAF ID = 101
Enc(Kstd, Kleaf-1)
Enc(Kleaf, Kn47)
Enc(Kleaf, Kn46)
⋮
Enc(Kleaf, Kn8)
EKB

FIG. 46


S4601
EXTRACTS KSTD BASED ON LEAF ID
Kstd
S4602
Kstd
DKB PROCESSING Enc(Kstd, Kleaf)
Kleaf
S4603
Kleaf
DKB PROCESSING Enc(Kleaf, Kn8)
Kn8
Kn8
S4604
EKB PROCESSING
Kroot
Kroot
S4605
EKB PROCESSING Enc(Kroot, KEK)
KEK
KEK
S4606
Enc(KEK, Kcon) DECODE PROCESSING
Kcon